Technical Disclosure Commons

Defensive Publications Series

October 2020

ON DEMAND BIT INDEX EXPLICIT REPLICATION FORWARDING FOR OPTIMIZED REPLICATION IN IOT NETWORKS

Mankamana Mishra

Pascal Thubert

Manmeet Singh

Anuj Budhiraja

ON DEMAND BIT INDEX EXPLICIT REPLICATION FORWARDING FOR OPTIMIZED REPLICATION IN IOT NETWORKS

AUTHORS:
Mankamana Mishra
Pascal Thubert
Manmeet Singh
Anuj Budhiraja

ABSTRACT

An industrial Internet of Things (IoT) deployment can potentially have thousands of devices. The need to perform, for example, a software upgrade on such devices presents a number of difficulties. While Bit Index Explicit Replication (BIER) offers a potential solution to some of those difficulties, with respect to low-cost IoT devices, utilizing BIER also presents various challenges. To address these challenges techniques are presented herein that facilitate the on-demand installation and uninstallation of a BIER state for the optimal replication of a multicast flow in support of, for example, software upgrades in an IoT domain, such as Routing Protocol for Low-Power and Lossy Networks (RPL) domain.

DETAILED DESCRIPTION

Various requirements of IoT devices may include, for example:

1. An absence of states (which internally requires memory to maintain different tables in hardware).
2. A need for providing an optimal way to deliver content from a server to different endpoints (e.g., 1:N, meaning that replication is expected).
3. A need for employing optimal replication.
4. A need to employ on-demand replication for control plane setup followed by removal once it is done.

Consider an illustrative industrial IoT deployment that is depicted in Figure 1, below.

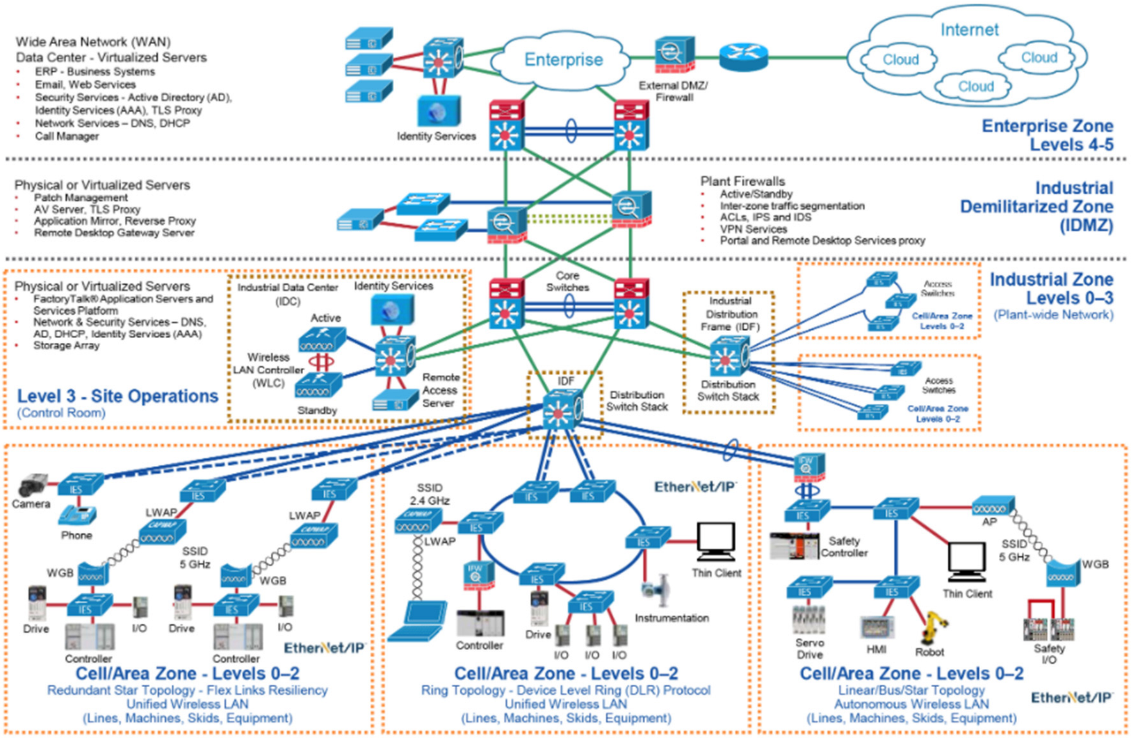


Figure 1: Illustrative Industrial IoT Deployment

As depicted in Figure 1, above, an industrial IoT deployment can potentially have thousands of devices. If there is a need for, as an example, software updates on thousands of such devices, it would be appropriate to use some form of point to multipoint technology, in other words some form of multicast technology.

However, various drawbacks of traditional multicast protocols may include, for example:

1. The use of each of protocol adds state in the network.
2. A periodic refresh is needed.
3. The necessary tree(s) need to be set up.

All of the above adds too much overhead and complexity in the network for low-cost IoT devices.

BIER provides for the optimal forwarding of multicast packets through a "multicast domain". However, it does not require a protocol for explicitly building multicast distribution trees, nor does it require intermediate nodes to maintain any per-flow state. When a multicast data packet enters the domain, the ingress router determines the set of egress routers to which the packet needs to be sent. The ingress router then encapsulates the packet in a BIER header. The BIER header contains a bit string in which each bit represents exactly one egress router in the domain. To forward the packet to a given set of egress routers, the bits corresponding to those routers are set in the BIER header.

With respect to low-cost IoT switches, one of the problems with BIER is the forwarding table. Each edge device needs to be assigned a bit. See, for example, Figure 2, below.

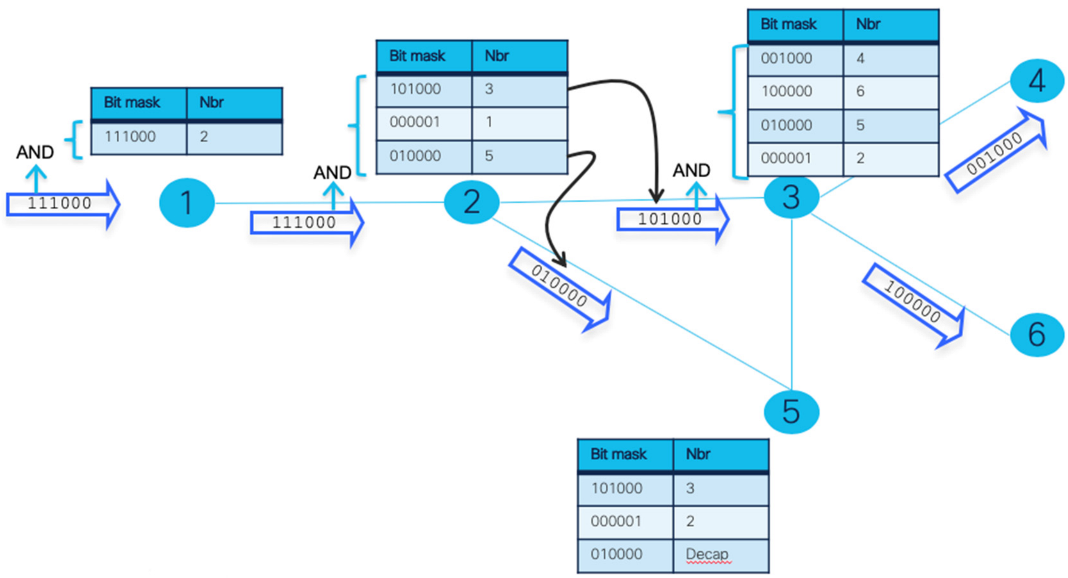
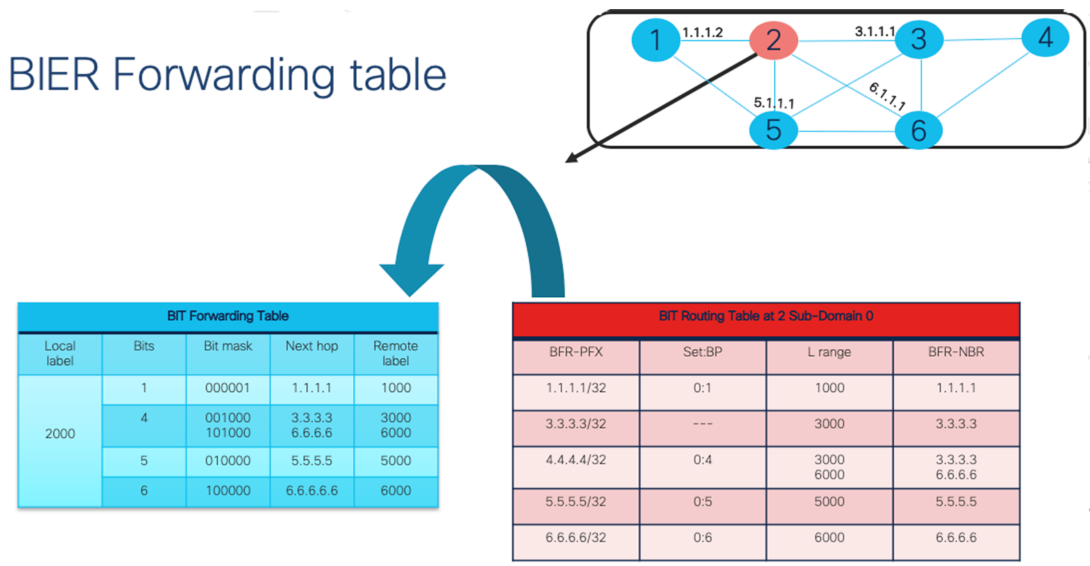


Figure 2: Illustrative Forwarding Tables

Thus, BIER provides a mechanism that supports optimal forwarding. However, if BIER is configured continually, it can add extra complexity of different forwarding tables that are present in switches, as shown in Figure 3, below, all of the time which may not be appropriate.



BIT Forwarding Table				
Local label	Bits	Bit mask	Next hop	Remote label
2000	1	000001	1.1.1.1	1000
	4	001000 101000	3.3.3.3 6.6.6.6	3000 6000
	5	010000	5.5.5.5	5000
	6	100000	6.6.6.6.6	6000

BIT Routing Table at 2 Sub-Domain 0			
BFR-PFX	Set:BP	L range	BFR-NBR
1.1.1.1/32	0:1	1000	1.1.1.1
3.3.3.3/32	---	3000	3.3.3.3
4.4.4.4/32	0:4	3000 6000	3.3.3.3 6.6.6.6
5.5.5.5/32	0:5	5000	5.5.5.5
6.6.6.6/32	0:6	6000	6.6.6.6

Figure 3: Illustrative BIER Forwarding Tables

For example, with just a small set of edge devices it is necessary to have these tables stored in all of the transit nodes. If thousands of switches are assigned bits in advance it would lead to having multiple forwarding tables all of the time.

Aspects of the techniques presented herein provide for the enhanced use of the BIER protocol to achieve optimal forwarding in the IoT network. At the same time the techniques support the removal of all of the extra forwarding table entries that would otherwise be present in the network.

Facets of particular interest and note within the techniques that are presented herein include, for example:

1. A server or controller assigns bits on demand. It may also associate a lifetime with a bit which means that after a maximum amount of time the bit would be released automatically.
2. Facet 1, above, establishes the replication network, and starts downloading the needed content.
3. The server starts forwarding the content.
4. Once the download is successful, the controller frees the assigned bits and assigns them to the next switch.

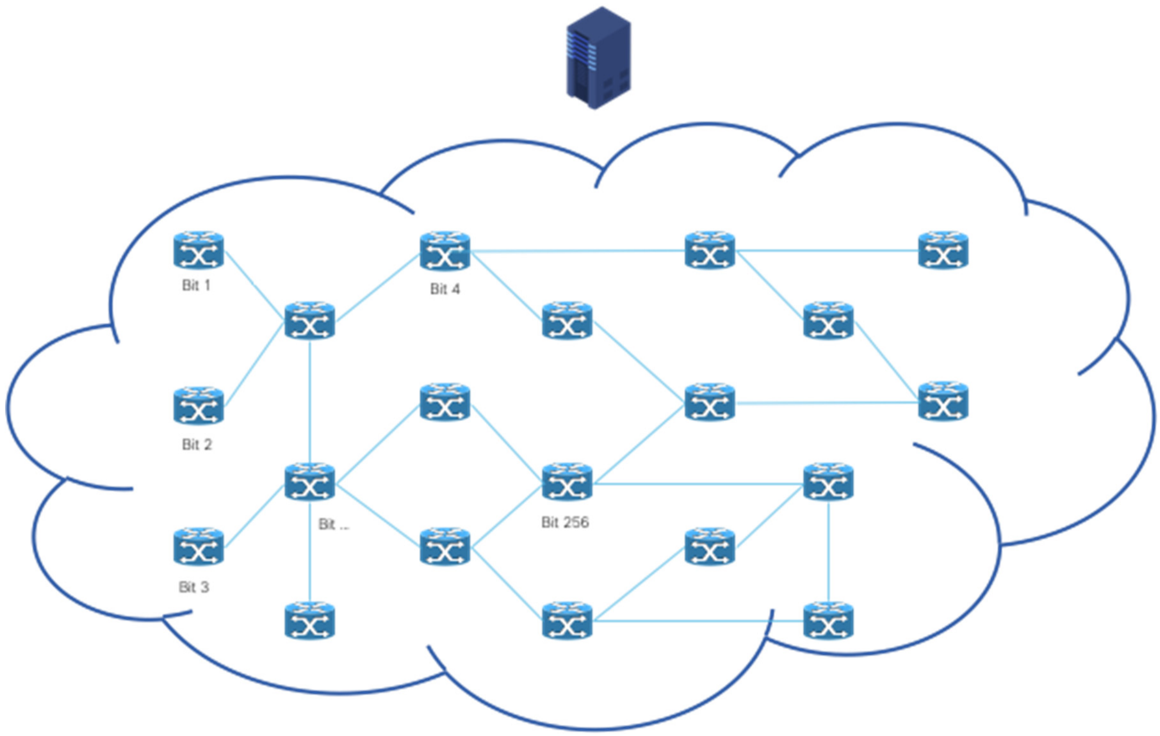


Figure 4: Exemplary Topology

Consider the exemplary topology that is depicted in Figure 4, above, in which a server is to download some content (e.g., a software upgrade). Depending upon the hardware capability the server assigns bits on the fly and starts downloading the software. Once the end devices have received the content, the controller or server removes the bit assignment and assigns it to another switch.

Aspects of the techniques presented herein may encompass the installation of a flow in which a first aspect may include the generation of a set of BIER destinations for a flow (e.g., for a software upgrade).

Such generation may use a variation of the diffusion algorithm to tell each of the destinations the Bit-Forwarding Router (BFR) ID for this multicast flow, and the particular bit that is assigned to that destination, with a sequence number that indicates the flow for which the BFR ID is used at this time. For that purpose, a novel message that contains type-length-value (TLV) artifacts for each destination is assembled, typically at the Bit-Forwarding Ingress Router (BFIR), that is the node that injects the BIER packet. Each

TLV artifact contains information such as the address of a destination and a bit associated to that destination in a BIER bitmap

A second aspect of the techniques presented herein may involve variation of a diffusion algorithm in support of message propagation. For example, the message as discussed above can be partitioned and each partition can be sent to one of the nodes listed in the partition. Recursively, the child nodes that receive a partition make a sub-partition of it and send each sub-partitioned piece to one of the nodes in that sub-partition piece, perhaps, as one example, the nearest node. This creates a distribution tree that is pseudo-random and is not necessarily the tree used later for BIER forwarding.

As with diffusion algorithms, when all of the children (if any) have acknowledged that they have handled their partition, and this parent has installed its own bit in the BIER routing, this parent can acknowledge to its own parent. When the BFIR has received an acknowledgment from all the nodes to which it sent a partition, it means that all of the destinations have received the bit information and injected it in their routing. If a child did not acknowledge, then the parent can retry with another child in the partition. It suffices that either of the children respond to ensure that the message was received by all the sub-partitions.

In one instances, techniques herein may also employ a RPL variation in address-less storing mode. In RPL BIER, the addresses of the nodes are not necessarily advertised in the Destination Advertisement Object (DAO). Only the bits are included. The message can be flooded in full through the RPL preferred parent tree or the full Destination Oriented Directed Acyclic Graph (DODAG). Each node reads it and recognizes whether it is listed. If it is, it retains the bit (i.e., a BFR ID) and it advertises the same in RPL as discussed in the Internet Engineering Task Force (IETF) Internet-Draft "RPL-BIER -- draft-thubert-roll-bier-02" (see https://tools.ietf.org/html/draft-thubert-roll-bier).

Note that under BIER, including RPL BIER, the routing protocol in place advertises the bit that corresponds to the destination with the destination, and the BIER forwarding follows the best path to each bit.

Once a flow is complete, a third aspect of techniques herein may involve employing a similar diffusion approach to remove the bits, which removes the BFR ID and all of the associated state form the routing.

It is important to note that aspects of the techniques that are presented herein support, among other things:

1. The dynamic generation of a group for, as an example, a selective firmware upgrade (e.g., BIER) or transient deterministic flows (e.g., BIER – Traffic Engineering (BIER-TE) and/or RAW, see the IETF Internet-Draft "BIER-TE extensions for Packet Replication and Elimination Function (PREF) and OAM -- draft-thubert-bier-replication-elimination-03" at https://tools.ietf.org/html/draft-thubert-bier-replication-elimination) structured as a set that can be partitioned and sub-partitioned.
2. A fast diffusion algorithm to program the BIER bits for BIER and BIER-TE, with care paid to the setup and installation of the bits and a particular version of that diffusion within RPL.

Aspects of the techniques that are presented herein pay particular attention to how to distribute and install on-demand BIER state for BIER (where a bit is a destination node) and BIER-TE (where a bit is a segment).

A normal multicast needs to install groups and rendezvous points statically before the group can be used. The association a node and a group is a very stable thing. This is advantageous for long-lived streams (e.g., television channels where all of the backbone is continuously flooded with a permanent flow). This is less practical for short-lived flows each time having a different destination set.

BIER and BIER-TE designate individually the destinations or segments for each flow. Arguably each packet could be its own flow with its own set of destinations or segments. If there are sufficient bits in the bitmap to signal every destination or segment then the bits may be set once and remain in the network and multicast packets for life.

However, the bitmap goes into every packet so its size is constrained. Size is even more constrained in the context of RPL as explained in the IETF Internet-Draft "RPL-BIER -- draft-thubert-roll-bier-02" (referenced above). Thus a mechanism is needed to selectively install the bits just for the duration of a flow. For example, consider a smart grid DODAG that contains 4K nodes. To distribute a new configuration file to 50 nodes that serve, for example, a particular building, arguably a 50-bit bitmap is sufficient. However, the bits need to be advertised by the new set of destinations and the previous set of bits need to be removed from the routing. Aspects of the techniques presented herein support precisely this. Note that with the IETF Internet-Draft "RPL-BIER -- draft-thubert-roll-bier-02" (referenced above) the Internet Protocol version 6 (IPv6) addresses of the 4K nodes are never injected in RPL. Only the bits are included. This means that only as many nodes as the bitmap size can receive packets at a given point in time. Aspects of the techniques presented herein decide which set of nodes need to be communicated with, unicast or as a group, at a given point of time (see element one, above) and then instructs them to inject the bits (see element two, the diffusion algorithm, above) after first uninstalling the previous set of bits (see element three, above).

Aspects of the techniques presented herein offer a number of benefits, including, for example:

1. Providing a mechanism to bring up on-demand an infrastructure for replication with very constrained signaling in the packets (BIER) suitable for IoT where the amount of state for routing and the frame size is constrained.
2. Having a deterministic hardware requirement. The number of bits can be changed depending on the segments for the network. For example, if one part of the network has devices which can support only up to 64 bits with respect to BIER forwarding then bit assignments would only occur in that range.
3. Removing the requirement for having any pre-setup of a tree or extra state.

In summary, techniques have been presented that support the on-demand installation and deinstallation of BIER state for the optimal replication of a multicast flow in support of, for example, software upgrades in an IoT (e.g., RPL) domain.